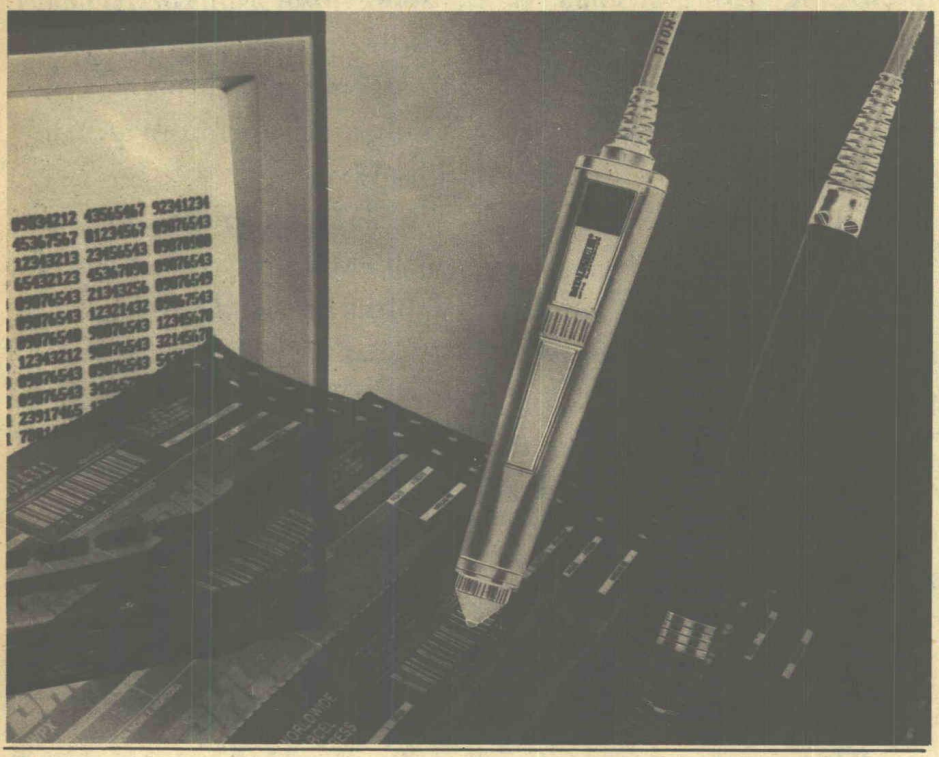
รหัสแถบ และโปรแกรมสำหรับพิมพ์รหัสแถบ



สมพัตร์ เบ็ญจชัยพร*

"เสนอแนวทางขั้นต้นและโปรแกรมใช้งานสำหรับรหัสแถบอย่างง่าย ๆ แต่ใช้งานได้อย่างมีประสิทธิภาพ"

ในปัจจุบันนี้มีผู้สั่งซื้ออุปกรณ์สำหรับใช้กับรหัสแถบ (BAR CODE) ทั้งเครื่องอ่านและเครื่องพิมพ์เข้ามาจำหน่ายภายในประเทศเป็นจำนวนมาก และแนวโน้มความต้องการใช้งานของกิจการต่าง ๆ ก็มากตามไปด้วย เนื่องจากความสะดวก ความเชื่อถือได้ของข้อมูล ความรวดเร็ว และปัจจัยสำคัญคือราคาไม่สูงมากนักเมื่อเทียบกับอุปกรณ์อื่น ๆ เป็นองค์ประกอบในการใช้งานนอกเหนือจากการใช้อย่างแพร่หลายและเป็นมาตรฐานทั่วโลก

นอกจากนี้ยังมีผู้สั่งซื้อโปรแกรมสำหรับการพิมพ์และอ่านเข้ามาจากต่างประเทศเพื่อทำให้การใช้งานสมบูรณ์ยิ่งขึ้น เมื่อมีผู้ให้ความสนใจในเรื่องนี้มากขึ้น ผู้เขียนจึงอยากเสนอแนวทางขั้นต้นและโปรแกรมใช้งานสำหรับรหัสแถบอย่างง่าย ๆ แต่ใช้งานได้อย่างมีประสิทธิภาพ เพื่อให้ผู้อ่านที่สนใจและต้องการใช้งานได้นำไปประยุกต์หรือดัดแปลงใช้งานเฉพาะด้านของตน

รหัสแถบกับการทำงานของคีย์บอร์ด

ผู้ที่เคยใช้งานไมโครคอมพิวเตอร์เพื่อป้อนข้อมูลในลักษณะ Data entry มักจะประสบปัญหาอย่างหนึ่งที่เหมือนกันคือ การเกิดข้อผิดพลาดเมื่อกดแป้นพิมพ์ผิดไป (Human error) ในบางครั้งเมื่อมีการป้อนข้อมูลจำนวนมาก ๆ และมีเวลาเพียงพอในการตรวจสอบก็อาจมีแนวทางแก้ไขข้อผิดพลาดได้ โดยการใช้คนตรวจสอบข้อมูลที่ป้อนเข้าไปทั้งหมดกับต้นฉบับ หรืออาจใช้บุคคล 2 คน ทำการป้อนข้อมูล 2 ครั้ง และใช้คอมพิวเตอร์ตรวจสอบซึ่งกันและกัน (Verification) ซึ่งแบบหลังเป็นที่นิยมมากในการทำงานเพราะมีความเชื่อถือได้ของข้อมูลในเกณฑ์สูง

แต่ในความเป็นจริงแล้วปัญหาบางอย่างไม่สามารถทำตามวิธีการนี้ได้ ตัวอย่างเช่น แคชเชียร์ของซูเปอร์มาร์เก็ตในห้างสรรพสินค้าขณะที่ทำการกดแป้นพิมพ์เพื่อคิดเงินของสินค้าให้กับลูกค้าที่มีเป็นจำนวนมาก จะมีโอกาสกดตัวเลขผิดพลาดได้มากเช่นกัน และการตรวจสอบตัวเลขจำนวนเงินที่ป้อนเข้าเครื่องในขณะนั้น ด้วยวิธีการข้างต้นย่อมกระทำได้ยากนอกจากการใช้ความรอบคอบเฉพาะตนเท่านั้น

จากการค้นคว้าวิจัยของฝ่ายวิจัยบริษัท Bell Telephone พบว่าความผิดพลาดจากการป้อนข้อมูลที่เป็นตัวเลขล้วน ๆ จะมีความผิดพลาดประมาณ

* ศูนย์วิจัยและพัฒนางานคอมพิวเตอร์ สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ

ร้อยละ 0.42–0.48 และผิดพลาดอย่างน้อย 1 ตัวอักษรต่อการป้อนข้อมูลทั่วไปแบบตัวอักษรปนตัวเลขจำนวน 208–230 ตัว ซึ่งสถิตินี้ได้ทดลองกับพนักงานป้อนข้อมูลที่มีความชำนาญมากอยู่แล้ว

เทคโนโลยีของรหัสแถบ จึงเข้ามาทดแทนเพื่อแก้ไขปัญหาที่เกิดขึ้นเนื่องจากอาศัยหลักการสะท้อนของแสงที่เรียกว่า Optical scanning โดยคอมพิวเตอร์จะอ่านข้อมูลได้โดยตรงไม่ผ่านเป็นพิมพ์ และข้อมูลที่เป็นรหัสจะแทนในรูปแบบของแถบขาวดำ (หรือสีอื่น) แบบแถบแคบ (Narrow bar) และแถบกว้าง (Wide bar) ซึ่งมีมาตรฐานสากลที่กำหนดเอาไว้ ความผิดพลาดที่ค้นพบจะมีประมาณ 1 ตัวอักษรต่อข้อมูลหลายล้านตัวอักษรเท่านั้น (ในสภาพที่เครื่องมือและอุปกรณ์มีความสมบูรณ์)

ความเป็นมาของรหัสแถบ

เริ่มเมื่อปี ค.ศ. 1949 สหรัฐอเมริกาได้ออกสิทธิบัตรรับรองรหัสแถบ แบบที่เรียกว่า Circular bar code ในปี ค.ศ. 1960 ก็มีการรับรองรหัสแถบแบบที่เรียกว่า Rail identification symbol หลังจากนั้นเป็นต้นมา เทคนิคของรหัสแถบรูปแบบต่าง ๆ ก็มีมากขึ้น และเริ่มใช้งานจริงจังเมื่อปี ค.ศ. 1970 เมื่อคณะกรรมการบริหารด้านห้างสรรพสินค้าของสหรัฐ–อเมริกา ได้นำรหัสที่เรียกว่า UPC (Universal Product Code) ซึ่งเป็นรหัสที่ใช้กันมากในสินค้า ออกเผยแพร่และใช้กันแพร่หลายในสหรัฐอเมริกา และยุโรป ตั้งแต่ ค.ศ. 1973 และ ค.ศ. 1977 ตามลำดับ

การใช้งานในด้านอื่น ๆ เริ่มตั้งแต่ปี ค.ศ. 1980 เป็นต้นมา เมื่อหน่วยป้องกันประเทศของสหรัฐอเมริกา ใช้เพื่อเช็คอุปกรณ์เครื่องมือและยุทโธปกรณ์คงคลัง ขณะที่โรงงานอุตสาหกรรมนำไปใช้เพื่องานต่าง ๆ มากขึ้น แต่คนทั่วไปเริ่มคุ้นเคยกับรหัสแถบเป็นอย่างดีจากรหัสสินค้า และการชำระเงินที่คอมพิวเตอร์รวมออกมาจากการอ่านรหัสแถบบนสินค้าเหล่านั้น จากความสะดวกเหล่านี้ทำให้สามารถลดพนักงาน ณ จุดนี้ลงได้ ในปี ค.ศ. 1981 มีห้างสรรพสินค้ามากกว่า 4000 แห่ง ในสหรัฐอเมริกาและแคนาดาใช้รหัสแถบในธุรกิจนี้ นอกจากนี้ยังใช้กับกิจการอื่น เช่น ห้องสมุด บริการสุขภาพ งานเอกสาร งานการทหาร การวิจัย การผลิตสินค้า เป็นต้น

รูปที่ 1
ตัวอย่างการใช้รหัสแถบบนสินค้า

หลักการของรหัสแถบ

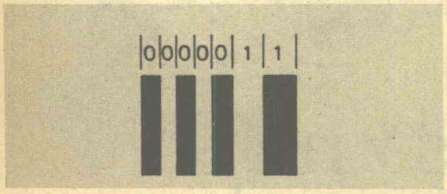
รหัสแถบเป็นการแทนข้อมูลที่เป็นรหัสของเลขฐานสอง (Binary code) ในรูปแบบของแถบขาว – ดำ ที่มีความกว้างของแถบที่ต่างกันในแถบดำหรือแถบขาวที่มีความกว้างจะแทนค่าเป็น 1 และแถบที่มีความแคบ (หรือมองด้วยตาเหมือนเป็นเส้นตรงเล็ก ๆ) ทั้งขาวและดำจะมีค่าเป็น 0 แถบขาว – ดำ ที่ว่านี้มีลักษณะและชื่อที่ใช้คือ

แถบสีดำที่มีความกว้างมาก เรียกว่า Wide bar

เส้นหรือแถบสีดำที่มีความกว้างน้อย เรียกว่า Narrow bar

ช่องว่าง (แถบสีขาว) ที่มีความกว้างมาก เรียกว่า Wide space

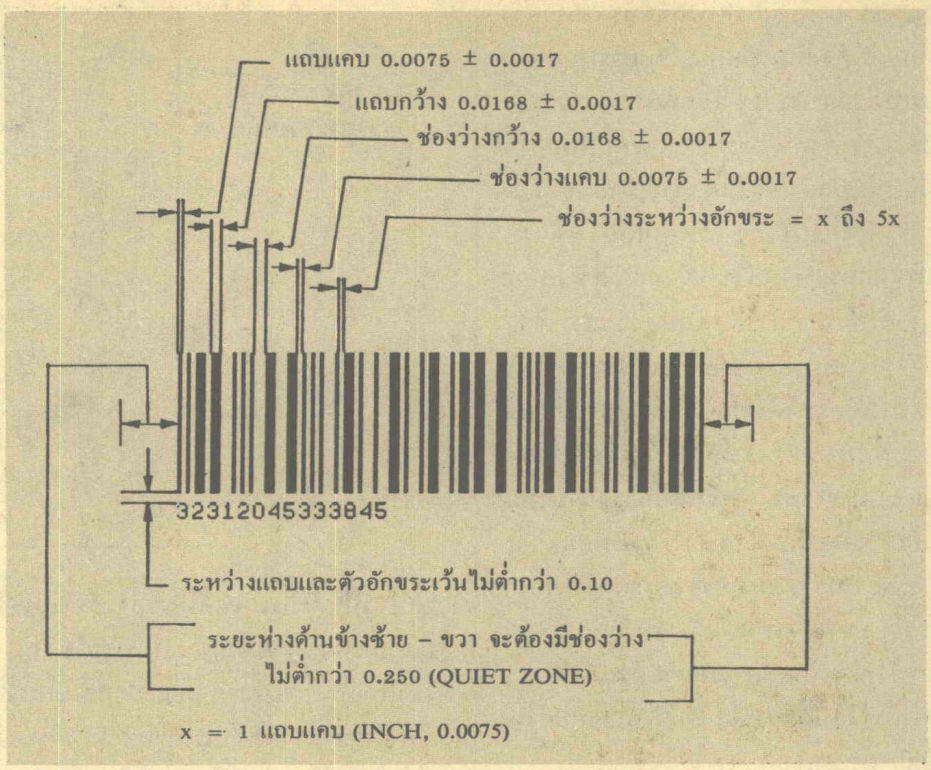
ช่องว่าง (เส้นสีขาว) ที่มีความกว้างน้อย เรียกว่า Narrow space



รูปที่ 2 การแทนค่าเลขฐานสองของแถบต่าง ๆ

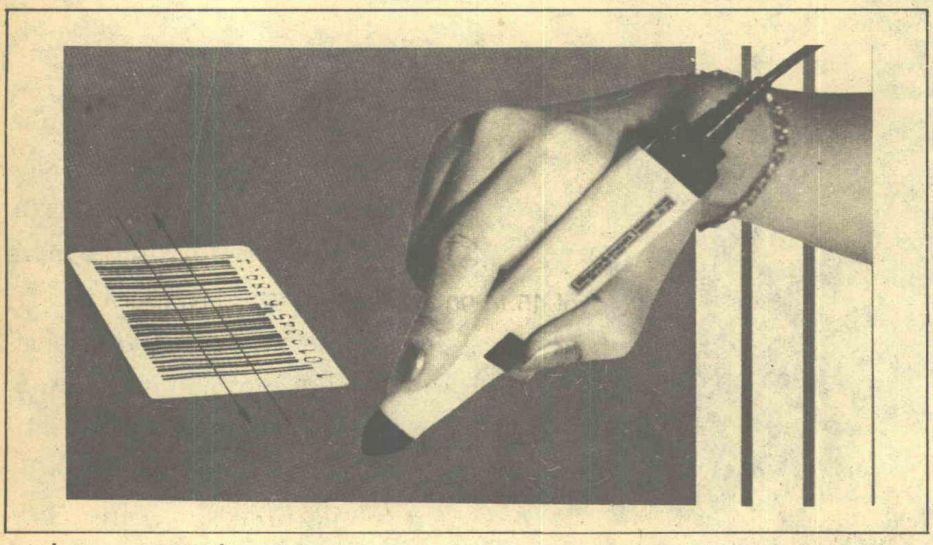
การอ่านรหัสแถบ

ในการอ่านรหัสแถบใช้หลักการเปลี่ยนรหัสแถบให้เป็นรหัสแอสกี้ โดยอาศัยความแตกต่างกันระหว่างแถบเข้มและพื้นที่ว่าง โดยที่พื้นที่ว่าง (ปกติจะเป็นสีขาวหรือสีอ่อน) จะมีการสะท้อนกลับของแสงได้มากกว่าบริเวณที่เป็นแถบเข้ม (ซึ่งใช้สีดำหรือสีอื่นที่มีความเข้มมาก) ตัวอ่าน (Bar code reader) จะประกอบด้วยตัวกำเนิดแสงที่ส่งผ่านเลนซ์ออกมาโดยถูกบังคับทิศทางให้มีจุดรวมแสงเล็กที่สุด กับตัวรับแสงที่มีความไวสูง ทั้ง 2 อย่างนี้จะบรรจุไว้ในตัวอ่านเดียวกันที่มีหลายรูปแบบ แต่แบบที่เป็นพื้นฐานที่สุดอยู่ในรูปคล้ายปากกาขนาดใหญ่ (Wand type)



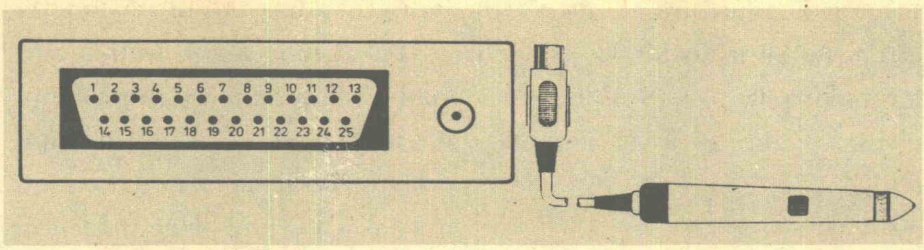
รูปที่ 3 ลักษณะและขนาดของรหัสแถบมาตรฐาน

ตัวอ่านจะถูกสแกนผ่านรหัสแถบในขณะที่ตัวกำเนิดแสงจะทำให้เกิดแสงส่งผ่านเลนซ์ไปกระทบบนรหัสแถบและสะท้อนกลับจากแถบ (แถบและช่องว่าง) กลับไปยังตัวรับแสง (Photosensor) ที่เกิดค่าความแตกต่างขึ้นตามหลักการสะท้อนกลับในแต่ละแถบ ทำให้เกิดสภาวะลอจิก "0" และลอจิก "1" ขึ้นตามที่กล่าวมาแล้วข้างต้น ซึ่งเมื่อรวมสภาวะลอจิก "0" และ "1" ทั้งหมดตลอดความกว้างของทุกแถบแล้วจะตรงกับแพตเทิร์นที่ได้กำหนดไว้แล้ว ในตัวอ่านรหัสแถบจะใช้ตัวกำเนิดแสงสีแดงหรือสีขาว แต่ส่วนใหญ่จะใช้แสงสีแดงเนื่องจากแสงสีขาวต้องการพลังงานและความเข้มของแสงสูงมากกว่าสีแดง แสงสีแดงสามารถอ่านรหัสที่พิมพ์ด้วยสีต่าง ๆ ได้ยกเว้นรหัสที่พิมพ์ด้วยสีแดง



รูปที่ 4 การใช้เครื่องอ่านสแกนตามแนวกว้างของรหัสแถบ

องค์ประกอบที่สำคัญ 2 ประการที่จำเป็นอย่างมากในการอ่านรหัสแถบได้ถูกต้อง ประการแรกคือ พื้นที่ภายในแถบและช่องว่าง จะต้องทำให้เกิดความแตกต่างของการสะท้อนกลับอย่างมาก (Contrast) เช่น แถบสีดำและช่องว่างสีขาว เป็นต้น ซึ่งปกติความแตกต่างนี้จะต้องอยู่ในช่วงระหว่างอัตรา 80–90 เปอร์เซ็นต์ขึ้นไป ประการสุดท้ายคือความกว้างระหว่างแถบกว้าง หรือช่องว่าง



รูปที่ 5 ลักษณะของเครื่องอ่านแบบที่ต่อกับ Com port ของไมโครฯ

กว้างต่อแถบแคบ หรือช่องว่างแคบจะเป็นอัตราส่วน 2 : 0.5, 2 : 1 และ 3 : 1

ตัวอ่านรหัสแถบแบบปากกาที่มีจำหน่ายในเมืองไทย ชนิดที่นำมาต่อกับไมโครคอมพิวเตอร์มี 2 แบบ ที่นิยมกันมากคือ แบบที่ต่อเข้ากับ RS422 หรือ RS232 (COM1, COM2) ซึ่งจะมีชุดควบคุม (Controller) ที่สามารถปรับค่าต่าง ๆ ได้ตามความต้องการ เช่น ความเร็ว ชนิดของรหัส ฯลฯ และอีกแบบหนึ่งจะใช้ต่อแทนคีย์บอร์ด โดยใช้ Keyboard emulator เป็นตัวควบคุมการทำงาน ทั้ง 2 แบบให้คุณสมบัติในการอ่านและเปลี่ยนรหัสได้ใกล้เคียงกันมาก และยังสามารถอ่านได้ทั้ง 2 ทิศทางโดยไม่ผิดพลาด นอกเหนือจากคุณสมบัติเฉพาะของเครื่องแล้วยังขึ้นอยู่กับโปรแกรมที่เขียนขึ้นมาควบคุมโดยผู้ใช้อีกด้วย

นอกจากนี้ยังมีตัวอ่านที่มีประสิทธิภาพสูงประเภทเลเซอร์ (Laser bar code scanner) ซึ่งมีทั้งชนิดที่ยังต้องใช้คน และแบบสแกนอัตโนมัติ ทั้ง 2 แบบนี้ให้ความรวดเร็ว ความเชื่อถือได้ และลดขนาดการพิมพ์รหัสแถบให้เล็กลงได้อย่างมาก

ชนิดของรหัสแถบ

ปัจจุบันชนิดของรหัสแถบที่นิยมใช้แพร่หลายแบ่งได้เป็น

1. ชนิดรหัส 2 ใน 5 (2 of 5 code)

เป็นรหัสที่มีใช้ตั้งแต่ ค.ศ. 1960 เป็นแบบที่ง่ายที่สุดในการใช้งาน การที่ชื่อเรียกว่า 2 ใน 5 เพราะใน 1 รหัสจะประกอบไปด้วยแถบ 5 แถบ (5 บิต) แต่จะมีแถบกว้างที่มีค่าเป็น 1 (แถบกว้าง) เพียง 2 แถบ (2 บิต) เท่านั้น ส่วนบิตที่เหลือเป็น 0 ทั้งหมด คือการแทนด้วยแถบแคบ (Narrow bar) 3 แถบ โดยไม่นำส่วนที่เป็นช่องว่าง (Space bar) มาใช้เลย

รหัส 2 ใน 5 นี้ เป็นรหัสที่ใช้แทนข้อมูลได้เฉพาะตัวเลข 0 – 9 เพียงแค่ 10 รหัสเท่านั้น เริ่มต้นจาก Start code 3 บิต คือ 110 (แถบกว้าง 2 และแถบแคบ 1) กับปิดท้ายด้วย Stop code 3 บิต คือ 101 ส่วนรหัสทั้ง 5 บิต ที่แทนเลข 0 – 9 ดูได้จากตารางในรหัส Interleaved 2 of 5

รูปที่ 6 รหัสแถบชนิด 2 ใน 5

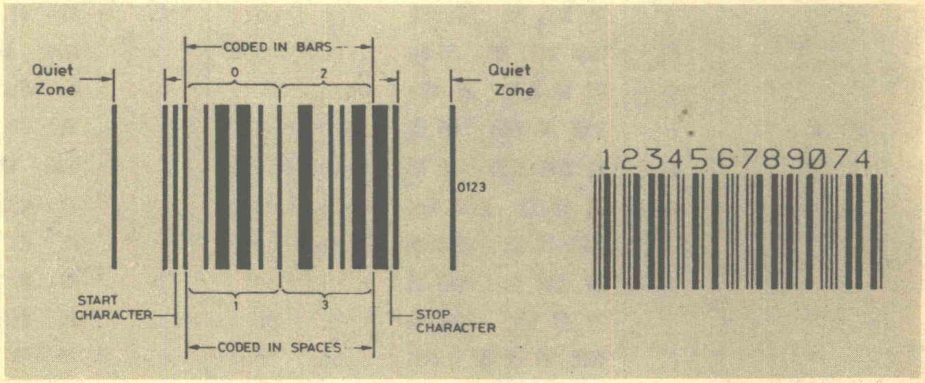
2. ชนิดรหัส 2 ใน 5 แบบสอดแทรก (Interleaved 2 of 5)

รหัสแบบนี้คล้ายคลึงกับแบบแรกมาก เพราะพัฒนามาจากรหัสแบบแรก เนื่องจากรหัส 2 ใน 5 ไม่ได้นำส่วนที่เป็นช่องว่างกว้าง และช่องว่างแคบมาใช้ คงใช้แต่เพียงแถบกว้างและแถบแคบ จึงทำให้ความหนาแน่นของข้อมูลน้อยลง นั่นคือ เมื่อต้องการบรรจุข้อมูลต่อเนื่องหลายตัวเลข จะต้องใช้แถบที่มีความกว้างมากขึ้น

รหัส 2 ใน 5 แบบสอดแทรก ได้ดัดแปลงนำส่วนที่เป็นช่องว่างทั้ง 2 ชนิดมาใช้งานด้วย โดยการสอดแทรกรหัสลงไปอีก 1 รหัส ทุก ๆ ช่วง 5 แถบของรหัสปกติที่เป็นแถบสีดำ แต่ก็ยังแทนรหัสตัวเลข 0 – 9 ได้เพียง 10 รหัสเท่านั้น

จากรูปที่ 7 ใช้แทนรหัสตัวเลข 1234... ตามลำดับ การใช้งานของรหัส 2 ใน 5 แบบสอดแทรกจะเริ่มต้นส่วนที่เป็น Start code ทางด้านซ้ายประกอบด้วยแถบแคบ 2 แถบ และช่องว่างแคบ 2 แถบสลับกัน ส่วนทางด้านขวาเป็น Stop code ประกอบด้วยแถบกว้าง 1 แถบ ช่องว่างแคบ 1 แถบ และแถบแคบ 1 แถบ ตามลำดับ

ภายในระหว่าง Start และ Stop code แบ่งเป็น 2 ส่วน ส่วนแรกคือ ส่วน



รูปที่ 7 รหัสแถบชนิด 2 ใน 5 แบบสอดแทรก

ที่เป็นแถบดำกว้างและแคบ จะใช้แทนรหัสเหมือน 2 ใน 5 ขณะเดียวกันในส่วนของแถบเหล่านี้จะมีแถบช่องว่างสีขาวกว้างและแคบแทนได้เช่นเดียวกับรหัส 2 ใน 5 ปกติ จากตัวอย่าง 5 แถบแรกที่เป็นสีดำแทนค่าได้เท่ากับ 1 แถบ

ตัวเลข	เลขฐาน 2 ทั้ง 5 แบบ				
0	0	0	1	1	0
1	1	0	0	0	1
2	0	1	0	0	1
3	1	1	0	0	0
4	0	0	1	0	1
5	1	0	1	0	0
6	0	1	1	0	0
7	0	0	0	1	1
8	1	0	0	1	0
9	0	1	0	1	0

รูปที่ 8 ตารางรหัสเลขฐาน 2 ของรหัส 2 ใน 5 ทั้งสองแบบ

ขาวในช่วงเดียวกันเท่ากับ 2 สีดำช่วงต่อมา 5 แถบแทนได้เท่ากับ 3 และ 5 แถบขาวต่อมาเท่ากับ 4 เช่นนี้ตลอดไปจนหมด รวมเป็นค่าที่อ่านได้เท่ากับ 1234.....

3. ชนิดรหัส 3 ใน 9 (3 of 9 or 39 code)

รหัส 3 ใน 9 เป็นรหัสที่ใช้แทนตัวอักขระทั้งหมด 44 อักขระ เป็นอักษรตัวใหญ่ 26 รหัส เลข 0–9 10 รหัส และอักขระพิเศษอีก 8 รหัส เป็นการประยุกต์ใช้รหัส 2 ใน 5 โดยการนำเอาส่วนที่เป็นแถบดำ 5 แถบ และแถบว่าง 4 แถบ รวมเป็น 9 แถบ แทน 1 รหัส ในแถบดำ 5 แถบนั้น ประกอบด้วยแถบกว้างที่เป็นบิต 1 อยู่ 2 แถบ และแถบแคบที่เป็นบิต 0 อยู่ 3 แถบ ส่วนแถบว่าง (space) 4 แถบ ประกอบด้วยแถบกว้างที่เป็นบิต 1 อยู่ 1 แถบ และแถบแคบที่เป็นบิต 0 อีก 3 แถบ ดังนั้นเมื่อรวมทั้งหมด 9 แถบ จะเป็นบิต 1 อยู่ 3 แถบ และบิต 0 อยู่ 6 แถบ

รหัส 3 ใน 9 มีส่วนเริ่มต้น (Start code) และสิ้นสุด (Stop code) ด้วยรหัสเดียวกันคือ * (Asterisk) ซึ่งมีรหัสฐาน 2 เป็นแถบ 00110 และช่องว่าง 1000

รูปที่ 9 รหัสแถบชนิดรหัส 3 ใน 9

รูปที่ 10 ตารางเลขฐาน 2 ของรหัส 3 ใน 9

อักขระ	แพตเทิร์น	แถบ ช่องว่าง	อักขระ	แพตเทิร์น	แถบ ช่องว่าง
1		10001 0100	M		11000 0001
2		01001 0100	N		00101 0001
3		11000 0100	O		10100 0001
4		00101 0100	P		01100 0001
5		10100 0100	Q		00011 0001
6		01100 0100	R		10010 0001
7		00011 0100	S		01010 0001
8		10010 0100	T		00110 0001
9		01010 0100	U		10001 1000
0		00110 0100	V		01001 1000
A		10001 0010	W		11000 1000
B		01001 0010	X		00101 1000
C		11000 0010	Y		10100 1000
D		00101 0010	Z		01100 1000
E		10100 0010	–		00011 1000
F		01100 0010			10010 1000
G		00011 0010	SPACE		01010 1000
H		10010 0010	*		00110 1000
I		01010 0010	$		00000 1110
J		00110 0010	/		00000 1101
K		10001 0001	+		00000 1011
L		01001 0001	%		00000 0111

ข้อดีของรหัสชนิดนี้คือใช้งานได้กว้าง-ขวางมากขึ้น เพราะสามารถใช้ตัวเลขปนกับตัวอักษรและเครื่องหมายต่าง ๆ ได้ ซึ่งรหัสทั้งหมดแทนได้ตามรูปที่ 10

4. ชนิดรหัส Codabar

รหัส Codabar ประกอบด้วย 7 บิต โดย 4 บิตเป็นแถบดำ และ 3 บิตเป็นช่องว่าง ใช้แทนตัวเลข 0 – 9 เครื่อง-หมาย – $: / . + A B C และ D

รหัส Codabar ที่สมบูรณ์จะต้องมีรหัสที่ใช้แทนตัวอักษร A B C หรือ D (เช่น A = 0011010) เป็นส่วนเริ่มต้นหรือสิ้นสุด ภายในประกอบด้วยรหัสของ Codabar ที่เป็นตัวเลขและเครื่องหมาย ซึ่งทำให้มีความยาวไม่แน่นอนเพราะ 12 รหัสแรกมีบิตที่เป็น 1 อยู่ 2 บิต 4 รหัสต่อมามีบิต 1 อยู่ 3 บิต (Codabar ใช้ทั้งแถบดำและขาวแทนข้อมูลใน 1 รหัส) และ 4 รหัสสุดท้ายเป็นรหัสของ A, B, C, D กำหนดขึ้นมาเพื่อใช้เป็นรหัสเริ่มต้นและสิ้นสุด (Start/Stop code)

5. ชนิดรหัส UPC หรือ EAN Code (Universal Product Code or European Article Numbering)

เป็นรหัสที่นิยมใช้กับสินค้าที่มาจากต่างประเทศหลายประเภท หรือสินค้าที่ส่งออกไปขายต่างประเทศ ผู้อ่านหลาย ๆ คนคงได้เคยเห็นรหัสชนิดนี้ปรากฏบนสินค้าที่ซื้อ โครงสร้างของรหัสชนิดนี้ต่างจากรหัสแถบประเภทอื่น ๆ โดยสิ้นเชิง

จากรูปจะเห็นได้ว่ารหัสแถบชนิดนี้แบ่งเป็น 2 ส่วน ซึ่งถูกแบ่งด้วยแถบสีดำเล็ก ๆ แต่ยาวกว่าแถบอื่น 2 แถบคั่นอยู่ตรงกลาง (เลขรหัสฐาน 2 ของแถบคั่นกลางนี้เป็น 01010) และยังมีแถบลักษณะเดียวกัน 2 ชุด อยู่ทางซ้าย–ขวาสุด (เลขรหัสฐาน 2 ของแถบนี้คือ 101) แถบทั้ง 3 ชุดนี้เรียกว่า Guide bar ซึ่งปกติจะมีความยาวกว่าแถบอื่น ๆ เป็นข้อสังเกตทำให้แบ่งรหัสแถบเป็น 2 ส่วน คือโซนทางซ้ายและโซนทางขวา หลักสุดท้ายทางซ้ายขวาสุด เป็นตัวตรวจสอบความถูกต้อง (check digit) ซึ่งคำนวณมาจากหลักที่เหลือ โดยตัวตรวจสอบทางซ้ายสุดมาจากเลข 5 หลักทางซ้าย และหลักทางขวามาจากเลข 5 หลักที่อยู่ทางด้านขวา ซึ่งแถบสำหรับตรวจสอบนี้บางครั้งก็พิมพ์ยาวเท่ากับส่วนที่เป็น guide bar

จากรหัสของ UPC รหัสทางซ้ายจะใช้กับรหัสแถบแบบ UPC ในโซนทาง

รูปที่ 11 รหัสแถบชนิด Codabar

รูปที่ 13 รหัสแถบชนิด UPC/EAN

ตัวเลข	รหัสโซนซ้าย	รหัสโซนขวา
0	0001101	1110010
1	0011001	1100110
2	0010011	1101100
3	0111101	1000010
4	0100011	1011100
5	0110001	1001110
6	0101111	1010000
7	0111011	1000100
8	0110111	1001000
9	0001011	1110100

รูปที่ 14 รหัสเลขฐาน 2 ของ UPC โซนทางซ้ายและโซนทางขวา

รูปที่ 12 ตารางเลขฐาน 2 ของรหัส Codabar

หมายเลข	แพตเทิร์น	รหัส 7 บิต	อักขระ	แพตเทิร์น	รหัส 7 บิต
0		0000011	–		0001100
1		0000110	$		0011000
2		0001001	:		1000101
3		1100000	/		1010001
4		0010010	.		1010100
5		1000010	+		0010101
6		0100001	A		0011010
7		0100100	B		0101001
8		0110000	C		0001011
9		1001000	D		0001110

ซ้าย ส่วนรหัสทางขวาจะใช้ได้กับโซนทางขวาของรหัสแถบชนิด UPC เท่านั้น จะใช้สลับกันไม่ได้ ในส่วนของรหัสทางซ้ายจะขึ้นต้นด้วยบิต 0 และลงท้ายด้วยบิต 1 เสมอ จะมีการตรวจสอบเป็นแบบบิตคี่ (odd parity) ส่วนรหัสทางขวาจะกลับกับรหัสทางซ้าย คือมีบิต 1 เป็นบิตเริ่มต้น และ 0 เป็นบิตสิ้นสุด การตรวจสอบบิตเป็นแบบคู่ (even parity) นอกจากนี้จากตารางเลขรหัสทางซ้ายและทางขวายังเป็นเลขแบบ 1's comprement ซึ่งกันและกัน

รหัส UPC/EAN มีหลายประเภท คือ UPC-A, UPC-B, UPC-C, UPC-D UPC-E, EAN-8 และ EAN-13 ซึ่งมีความแตกต่างกันไปบ้าง การอ่านรหัสชนิดนี้จะยากกว่าแบบอื่น ถึงแม้ว่าแถบดำมีค่าเป็นบิต 1 และแถบขาวมีค่าเป็นบิต 0 เหมือนแบบอื่น ๆ ก็ตาม เนื่องจากในแถบดำ–ขาว ที่ใช้ไม่ได้มีแค่แถบกว้าง, ช่องว่างกว้าง, แถบแคบ หรือช่องว่างแคบเท่านั้น ในแถบดำ (bar) และแถบขาว (space) ยังแบ่งอย่างละ 4 ขนาด คือ แถบดำแคบสุด มีค่า 1 ขนาดที่ 2 กว้างกว่าขนาดแคบสุดเล็กน้อยมีค่า 11 และขนาดที่ 3 มีค่า 111 ส่วนแถบกว้างสุดมีค่า 1111 ทำนองเดียวกันแถบขาว 4 ขนาดมีค่า 0, 00, 000 และ 0000 ตามลำดับ แต่ละรหัสตัวเลขจะประกอบด้วยแถบขาว–ดำ อย่างละ 2 แถบ

นอกจากนี้ยังมีรหัสแถบชนิดอื่นที่พัฒนาขึ้นมาอีกเพื่องานเฉพาะด้านในแต่ละกิจการ เช่น รหัส 11, รหัส Plessley, รหัส AMES, รหัส Nixdorf, รหัส Jan, รหัส 128 Subset B, รหัส Subset C เป็นต้น แต่อย่างไรก็ตาม ลักษณะของรหัสแถบที่ดี ควรมีคุณสมบัติดังนี้

- สามารถตรวจสอบความถูกต้องภายในรหัสได้
- ความกว้างและจำนวนของแถบต่อรหัสควรจะคงที่
- สามารถใช้แทนตัวเลขหรือตัวเลขปนตัวอักษรได้ครบ
- มีโครงสร้างแบบง่าย ๆ
- การอ่านด้วยความเร็วที่ต่างกันควรได้ค่าที่ถูกต้องเสมอ
- มีความหนาแน่นของข้อมูลต่อความกว้างของแถบสูง

การพิมพ์รหัสแถบ

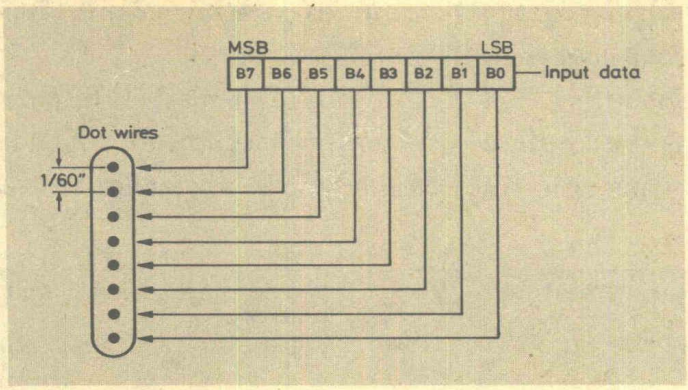
หลังจากทำความรู้จักกับรหัสแถบมาพอสมควรแล้วจะเห็นได้ว่า เราสามารถเลือกชนิดของรหัสแถบมาใช้งานตามความต้องการได้ทุกแบบ แต่จะมีแบบ UPC เท่านั้น ที่ค่อนข้างยุ่งยาก เพราะนอกจากคุณสมบัติเฉพาะแล้ว ยังต้องคำนึงถึงการใช้รหัสตัวเลขแทนค่าด้วย เช่น รหัสประเทศผู้ผลิต รหัสผู้ผลิตและรหัสประเภทสินค้า ที่ได้กำหนดไว้คงที่เป็นมาตรฐานทั่วโลก ซึ่งผู้ใช้จะต้องศึกษาให้เข้าใจก่อนนำมาใช้งาน ส่วน Codabar มีความกว้างของรหัส เป็น 2 แบบ และรหัสทั้งหมดก็สามารถใช้แบบ 3 of 9 แทนได้

ดังนั้นเพื่อตัดปัญหาที่ยุ่งยาก เราจึงจะนำชนิดที่เหลือและง่ายต่อการใช้งาน เช่น Interleaved 2 of 5 และ 3 of 9 มากำหนดโครงสร้างเพื่อเขียนโปรแกรมพิมพ์ โดยดูคุณสมบัติได้จากตารางรหัสแถบ ANSI

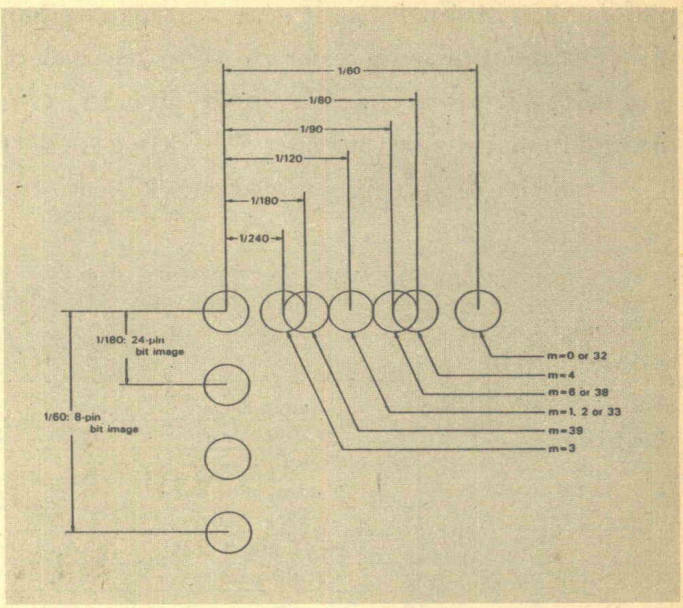
รูปที่ 15 ตารางรหัสแถบ ANSI

BAR CODE	ELEMENTS	ELEMENT WIDTH TOLERANCE T	WIDE-TO-NARROW ELEMENT RATIO N*
INTERLEAVED 2-OF-5	NARROW BAR W NARROW SPACE W WIDE BAR WIDE SPACE	$= \left(\frac{18N-21}{80}\right) W$	2:1 TO 3:1 (MUST EXCEED 2.2:1 WHENEVER NARROW ELEMENT < 0.02-1N, WIDE)
3-OF-9	SAME AS ABOVE	$= \frac{4}{27}(N-2/3)W$	SAME AS ABOVE
CODABAR	9 BAR WIDTHS 10 SPACE WIDTHS	$= \frac{0.0015}{0.0065} \times \frac{\text{ELEMENT}}{\text{WIDTH}}$	DOES NOT APPLY

N = THE RATIO OF THE WIDTH OF THE WIDE ELEMENT TO THE WIDTH OF THE NARROW ELEMENT (NOMINAL RATIO : N MUST BE HELD CONSTANT WITHIN AN INTERLEAVED 2 OF 5 AND 3 OF 9 BAR CODE SYMBOL).
FOR ALL THE ABOVE BAR CODES:
BAR CODE HEIGHT MINIMUM IS 0.25 IN. FOR HAND SCANNING OR 15% OF THE BAR CODE LENGTH. WHICHEVER IS GREATER: MINIMUM OF 1.25 IN. OR 25% OF THE BAR CODE LENGTH. WHICHEVER IS GREATER. FOR TRANSPORT PACKAGES.
MINIMUM NOMINAL WIDTH OF NARROW ELEMENTS IS 0.0075 IN. EXCEPT FOR DIRECT PRINTING ON CORRUGATED CONTAINERS. WHERE 0.040 IN. IS REQUIRED.
VOIDS OR SPOTS MEETING EITHER OF THE FOLLOWING ARE PERMITTED:
(1) CONTAINED WITHIN A CIRCLE WHOSE DIAMETER IS 0.4 TIMES THE NOMINAL WIDTH OF THE NARROW ELEMENT.
(2) OCCUPIES NO MORE THAN 25% OF THE AREA OF A CIRCLE WHOSE DIAMETER IS 0.8 TIMES THE NOMINAL WIDTH OF THE NARROW ELEMENT.
MINIMUM PRINT CONTRAST SIGNAL IS 75% IN THE B633 SPECTRAL BAND.



รูปที่ 16 ▲
ความสัมพันธ์ของข้อมูลแต่ละบิต กับเข็มพิมพ์



รูปที่ 17 ▶
ระยะห่างระหว่างเข็มพิมพ์แต่ละเข็มของเครื่องพิมพ์

โปรแกรมสำหรับพิมพ์รหัสแถบ ▶

Bit image mode สำหรับเครื่องพิมพ์แบบดอตแมตริกซ์

ตามปกติเครื่องพิมพ์แบบดอต-แมตริกซ์ทั่ว ๆ ไปมีคำสั่งควบคุมพิเศษอยู่ในการสั่งงานให้เครื่องพิมพ์แบบกราฟิก ที่เราเรียกว่า Bit image mode ความละเอียดที่ได้ขึ้นอยู่กับขนาดเส้นผ่าศูนย์กลางของเข็มพิมพ์ และจำนวนเข็มในการพิมพ์รหัสแถบนี้ได้ทดลองใช้เครื่องพิมพ์ของเอปซอน รุ่น LQ1050, LQ1500 และ LQ2500+ พิมพ์แบบ Bit image mode ซึ่งให้ผลออกมาใกล้เคียงกันมาก

การใช้คำสั่งเพื่อให้เครื่องพิมพ์แบบ Bit image จะต้องหาค่า N1, N2 จากจำนวนข้อมูลทั้งหมด โดยที่ค่าของ N1, N2 หาได้จาก

N1 = จำนวนเต็มของ (จำนวนข้อมูล/256)

N2 = (จำนวนข้อมูล/256 − N1) *256

จำนวนไบต์ของข้อมูลที่ส่งไปยังเครื่องพิมพ์เพื่อควบคุมเข็มพิมพ์ในแบบ

```
'***********************************
'**    BAR CODE PRINTING PROGRAM    **
'**           VERSION 1.0           **
'**              by                 **
'**      COMPUTER R&D CENTER        **
'**            KMIT-NB              **
'**            MARCH.88             **
'***********************************
DEFINT A-Z
DIM R(255),D$(59),E$(10)
'**** load binary code (3 of 9) ****
FOR I=1 TO 59
    READ D$(I)
NEXT I
'**** 3 of 9 data ****
DATA "011000100",,,,"010101000","000101010",,,,,"010010100"
DATA "010001010",,,"010000101","110000100","010100010"
DATA "000110100","100100001","001100001","101100000"
DATA "000110001","100110000","001110000","000100101"
DATA "100100100","001100100",,,,,,,,"100001001","001001001"
DATA "101001000","000011001","100011000","001011000"
DATA "000001101","100001100","001001100","000011100"
DATA "100000011","001000011","101000010","000010011"
DATA "100010010","001010010","000000111","100000110"
DATA "001000110","000010110","110000001","011000001"
DATA "111000000","010010001","110010000","011010000"
'**** load binary code (interlieaved 2 of 5) ****
FOR I = 1 TO 10
    READ E$(I)
NEXT I
'**** interleaved 2 of 5 data ****
DATA "00110","10001","01001","11000","00101"
DATA "10100","01100","00011","10010","01010"
'**** start program bar code printing ****
CLS
LOCATE 5,22 : PRINT "***********************************"
LOCATE 6,22 : PRINT "**    BAR CODE PRINTING PROGRAM    **"
LOCATE 7,22 : PRINT "**           VERSION 1.0           **"
LOCATE 8,22 : PRINT "**              by                 **"
LOCATE 9,22 : PRINT "**      COMPUTER R&D CENTER        **"
LOCATE 10,22: PRINT "**            KMIT-NB              **"
LOCATE 11,22: PRINT "**            MARCH.88             **"
LOCATE 12,22: PRINT "***********************************"
LOCATE 14,22: PRINT " PRESS ANY KEY TO CONTINUE ......  ";
Q$ = INPUT$(1)
'**** open printer file ****
CLS
LOCATE 10,20 : PRINT "       PLEASE CHECK YOUR PRINTER"
LOCATE 12,20 : PRINT "IF PRINTER ON LINE ALREADY, PRESS ANY KEY ...."
Q$ = INPUT$(1)
OPEN "O",1,"PRN"
100 '**** select bar width (standard and compact format) ****
CLS
LOCATE 5,25 : PRINT " SELECT BAR WIDTH"
LOCATE 7,25 : PRINT "1) STANDARD FORMAT"
```

▶

Bit image จะเท่ากับ N2×256+N1 สำหรับการเลือกพิมพ์โดยใช้ 8 เข็ม จากข้อมูลที่จะส่งในแต่ละไบต์จะมีความสัมพันธ์กับเข็มพิมพ์แต่ละเข็มดังนี้

จากคู่มือการใช้เครื่องพิมพ์เอปซอน เราสามารถเลือกใช้การส่งคำสั่งควบคุม (escape sequence) ใน Bit image mode โดยการส่ง CHR$(27); "*"; CHR$(M); CHR$(N1); CHR$(N2) หรือค่า ESC * ค่า N1, N2 หาได้จากสมการที่ผ่านมา ค่า M สามารถเลือกใช้ได้จากตารางในรูปที่ 18

จากค่าที่ได้ทดลองใช้ในโปรแกรมได้เลือกโหมดการพิมพ์เป็น double density แบบ 8 เข็ม โดยมีค่า M = 1

รูปที่ 18
ตารางเลือกค่า M
สำหรับการพิมพ์แบบ
Bit image

	m	Number of pins	m	Number of pins	Number of dots/inch	Number of dots/line (horizontal)	Inch/ second	Half dot
Normal-density	0	8	32	24	60	816	20	
Double-density	1	8	33	24	120	1632	10	
Double-speed, Double-density	2	8			120	1632	20	0
Quadruple-density	3	8			240	3264	10	0
CRT graphics	4	8			80	1088	10	
CRT graphics II	6	8	38	24	90	12	13.3	
Triple-density			39	24	180	2448	6.7	

```
    LOCATE 9,25 : PRINT "2) COMPACT FORMAT"
    LOCATE 12,25: PRINT "SELECT 1 OR 2 : ";
    Q$ = INPUT$(1)
    IF Q$ = "1" THEN
       C1 = 3
       C2 = 4
    ELSEIF Q$ = "2" THEN
       C1 = 2
       C2 = 3
    ELSE
       PRINT CHR$(7)
       LOCATE 14,22: PRINT "INVALID NUMBER, PRESS ANY KEY.....";
       Q$ = INPUT$(1)
       GOTO 100
    END IF
200 '**** select bar height ****
    CLS
    LOCATE 5,25 : PRINT "SELECT BAR HEIGHT"
    LOCATE 7,25 : PRINT "1) 1.5 CM.
    LOCATE 9,25 : PRINT "2) 2.0 CM.
    LOCATE 11,25: PRINT "3) 2.5 CM.
    LOCATE 14,25: PRINT "SELECT 1, 2 OR 3 : ";
    X$ = INPUT$(1)
    IF X$ = "1" THEN
       M = 18
    ELSEIF X$ = "2" THEN
       M = 25
    ELSEIF X$ = "3" THEN
       M = 32
    ELSE
       PRINT CHR$(7)
       LOCATE 16,25 : PRINT "INVALID NUMBER, PRESS ANY KEY ......";
       X$ = INPUT$(1)
       GOTO 200
    END IF
250 '**** select bar type ****
    CLS
    LOCATE 5,25 : PRINT " SELECT BAR CODE TYPE"
    LOCATE 7,25 : PRINT "1) INTERLEAVED 2 OF 5"
    LOCATE 9,25 : PRINT "2) 3 OF 9"
    LOCATE 12,25: PRINT " SELECT 1 OR 2 : ";
    P$ = INPUT$(1)
275 IF P$ = "1" THEN
       GOSUB 300
    ELSEIF P$ = "2" THEN
       GOSUB 400
    ELSE
       PRINT CHR$(7)
       LOCATE 16,25 : PRINT "INVALID NUMBER, PRESS ANY KEY ......";
       P$ = INPUT$(1)
       GOTO 250
    END IF
    CLS
    LOCATE 5,25 : PRINT "MORE DATA ? <Y/N> : ";
    Y$ = INPUT$(1)
```

ผู้ที่ไม่สะดวกในการใช้อาจเลือกใช้ คำสั่ง ESC L ได้โดยตรงซึ่งจะให้ผลแบบเดียวกันคือ CHR$(27); "L"; CHR$(N1); CHR$(N2) ซึ่งผลลัพธ์ที่ได้จะได้ความละเอียดต่าง ๆ ใกล้เคียงกับมาตรฐานมาก

นอกจากการพิมพ์แบบ Bit image mode แล้ว คำสั่งควบคุมที่สำคัญอีกอย่างหนึ่งคือ ระยะห่างระหว่างบรรทัด (line spacing) ซึ่งปกติในการพิมพ์แบบ Text mode มาตรฐานระยะห่างจะเท่ากับ 1/6 นิ้ว แต่ในการพิมพ์รหัสแถบจำเป็นต้องพิมพ์ Bit image ต่อกันเป็นเส้นตรง ระยะห่างที่เหมาะสมจากการทดลองคือเลือกใช้คำสั่งควบคุม 1/180 นิ้ว หรือเท่ากับ ESC 3 การพิมพ์ใช้การสั่งคำสั่งควบคุมเป็น CHR$(27); "3"; CHR$(N) โดยที่ค่า N อยู่ระหว่าง 0–255

โปรแกรมที่ใช้สำหรับพิมพ์รหัสแถบนี้ได้เขียนขึ้นเพื่อใช้กับเครื่องไมโครคอมพิวเตอร์ 16 บิต XT คอมแพตทิเบิ้ล และเครื่องพิมพ์เอปซอน (ผู้ที่มีเครื่องพิมพ์ยี่ห้ออื่น หรือเอปซอนรุ่นอื่นที่มีข้อกำหนดต่างจากนี้ สามารถใช้ได้โดยการเปลี่ยนคำสั่งควบคุมการพิมพ์ใน Bit image และควบคุม line spacing) ซึ่งเหมาะกับผู้ที่ต้องการนำไปประยุกต์ใช้ในงานของตน หรือสำหรับผู้ที่ต้องการ

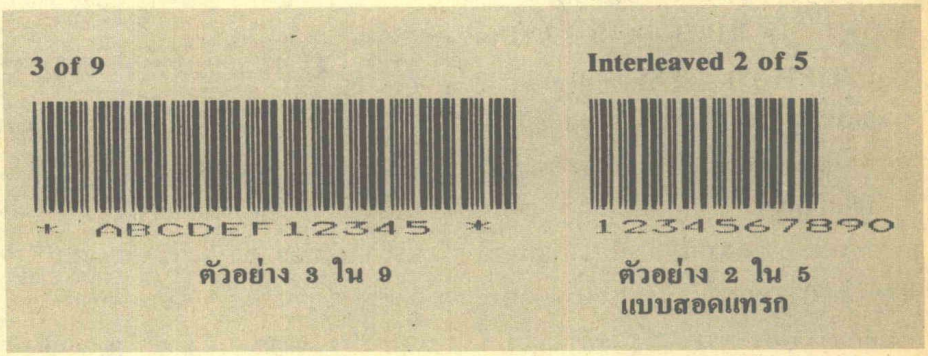
ทดลองพิมพ์รหัสแถบโดยไม่ต้องหาอุปกรณ์อื่นเพิ่มเติม

โปรแกรมสำหรับพิมพ์รหัสแถบ

โปรแกรมเขียนขึ้นจากเทอร์โบเบสิก ผู้ใช้สามารถเลือกชนิดของรหัสแถบได้ 2 แบบ คือ Interleaved 2 of 5 และ 3 of 9 นอกจากนี้ยังเลือกพิมพ์ได้ 2 ประเภท คือ ความกว้างของแถบขนาดมาตรฐาน และขนาดเล็กดังรูปที่ 19

เลขฐานสอง ที่อ่านด้วยคำสั่ง READ และ DATA เป็นเลขแทนรหัสของรหัสแถบแต่ละชนิด โดยให้ 1 บิต แทนความหมายของแถบกว้าง, ช่องว่างกว้าง, แถบแคบ และช่องว่างแคบ

คำสั่ง FOR M1 = 1 TO M เป็นจำนวนครั้งในการพิมพ์รหัสแถบ 1



รูปที่ 19 ตัวอย่างผลลัพธ์ที่ได้จากการพิมพ์ของโปรแกรม

(ต่อ)

```
    IF Y$ = "y" OR Y$ = "Y" THEN 275
    LOCATE 7,25 : PRINT "PRINT ANOTHER TYPE ? <Y/N> : ";
    Z$ = INPUT$(1)
    IF Z$ = "y" OR Z$ = "Y" THEN 100
    CLOSE
    LOCATE 9,25 : PRINT "END PROGRAM......."
    END
300 '**** input data sequence for interleaved 2 of 5 ****"
    Q$ = ""
    CLS
    LOCATE 5,5  : PRINT "INPUT DATA SEQUENCE OF INTERLEAVED 2 OF 5 <NUMERIC 0-9 ONLY>"
    LOCATE 7,5  : PRINT "NOT LONGER THAN 11 DIGITS AND NOT LESS THAN 2 DIGITS"
    LOCATE 9,5  : INPUT "ENTER DATA SEQUENCE : ";Q$
    IF LEN(Q$) > 11 THEN
        PRINT CHR$(7)
        LOCATE 11,5 : PRINT "DATA TO LONG, REENTER"
        LOCATE 13,5 : PRINT "PRESS ANY KEY .......";
        Q$ = INPUT$(1)
        GOTO 300
    END IF
    IF LEN(Q$) < 2 THEN
        PRINT CHR$(7)
        LOCATE 11,5 : PRINT "DATA LESS THAN 2 DIGITS, REENTER"
        LOCATE 13,5 : PRINT "PRESS ANY KEY .......";
        Q$ = INPUT$(1)
        GOTO 300
    END IF
    FOR I = 1 TO LEN(Q$)
        IF MID$(Q$,I,1) < "0" OR MID$(Q$,I,1) > "9" THEN
            PRINT CHR$(7)
            LOCATE 11,5 : PRINT "INVALID DATA, REENTER"
            LOCATE 13,5 : PRINT "PRESS ANY KEY .......";
            Q$ = INPUT$(1)
            GOTO 300
        END IF
    NEXT I
    T$ = ""
    S$ = ""
    LOCATE 11,5 : PRINT "INTERLEAVED 2 OF 5 BAR CODE"
    FOR I = 1 TO LEN(Q$) STEP 2
        T$ = T$+E$(ASC(MID$(Q$,I,1))-47)
    NEXT I
    FOR I = 2 TO LEN(Q$) STEP 2
        S$ = S$+E$(ASC(MID$(Q$,I,1))-47)
    NEXT I
    R$ = ""
    FOR I = 1 TO LEN(T$)
        R$ = R$+MID$(T$,I,1)+MID$(S$,I,1)
    NEXT I
    R$ = "0000"+R$+"100"
    FOR I = 1 TO LEN(R$)
        R(I) = VAL(MID$(R$,I,1))
    NEXT I
    T$=R$
    PRINT #1, "INTERLEAVED 2 OF 5"
    PRINT #1,
    GOSUB 600
    RETURN
```

ชุด สูง 1.5, 2.0, 2.5 ซม. ระยะห่างระหว่างบรรทัดของการพิมพ์แต่ละครั้งถูกควบคุมด้วยคำสั่ง PRINT # 1, CHR$(27); "3"; CHR(5); ขณะที่คำสั่ง PRINT # 1, CHR$(27); "*"; CHR$(1); CHR$(N8); CHR$(N4); เป็นการควบคุมการพิมพ์ Bit image mode แบบ 8 เข็ม double density ในการพิมพ์แต่ละครั้งจะส่งคำสั่งควบคุมเพื่อยกเลิกโหมดการพิมพ์พิเศษต่าง ๆ เป็นโหมดปกติโดยใช้ CHR(27); "@" (Initialize printer) ผู้อ่านอาจสงสัยว่าทำไมไม่ใช้คำสั่ง LPRINT เพื่อสั่งออกเครื่องพิมพ์โดยตรง แทนการเปิดไฟล์ PRN หรือ LPT1 ที่ใช้คำสั่งนี้แทนเพราะการใช้ LPRINT จะให้ผลลัพธ์ทางเครื่องพิมพ์ผิดพลาดมากกว่าการเปิดไฟล์เครื่องพิมพ์

บทสรุปและข้อเสนอแนะ

การใช้งานของโปรแกรมนี้ผู้เขียนได้ทดลองใช้งานดูแล้ว ได้ผลเป็นที่น่าพอใจเมื่อเทียบกับโปรแกรมที่มาจากต่างประเทศที่มีอยู่ อย่างไรก็ดีเนื่องจากปัญหาสำคัญทางการพิมพ์ของเครื่องพิมพ์ดอต-แมตริกซ์ในแบบ Bit image คือความเร็วไม่สูงนัก การพิมพ์ด้วยโปรแกรมนี้จึงไม่เหมาะสำหรับผู้ที่ต้องการพิมพ์ข้อมูลจำนวนมาก ๆ ในระยะเวลาจำกัด ซึ่งอาจแก้ไขโดยการใช้เครื่องพิมพ์แบบความร้อน (Thermal printer) สำหรับแถบ และ label ที่มีขายอยู่แต่ราคาค่อนข้างสูง อยู่ระหว่างหลักหมื่นถึงหลายแสนบาท หรืออาจใช้เครื่องพิมพ์แบบดอตแมตริกซ์ชนิด line printer ที่ให้ความเร็วสูงกว่าแบบดอตแมตริกซ์ธรรมดามาก (สำหรับโปรแกรมที่พิมพ์ด้วยเครื่องดอตแมตริกซ์ชนิด line printer ทางศูนย์วิจัยและพัฒนางานคอมพิวเตอร์ ได้ทดลองเขียนขึ้นมาด้วยภาษา C โดยใช้เครื่องซูเปอร์ไมโครคอมพิวเตอร์ที่ใช้ UNIX OS ได้ผลเป็นที่น่าพอใจมากทั้งด้านความถูกต้อง และความรวดเร็ว ซึ่งถ้ามีโอกาสจะนำมาเสนอต่อไป)

สำหรับผู้อ่านที่มีปัญหาไม่มีเทอร์โบเบสิก สามารถใช้ BASICA หรือ BASIC อื่นแทนได้โดยการเติมเลขบรรทัด (line number) ในทุก ๆ บรรทัด เนื่องจากคำสั่งที่ใช้ได้พยายามเลือกคำสั่งให้สามารถใช้กับ BASIC ทั่วไป แต่ข้อดีของเทอร์โบเบสิก คือความเร็วในการทำงานจะเร็วกว่า BASICA จนสังเกตเห็นได้ชัดเจน ข้อแนะนำอีกประการหนึ่งคือ โดยปกติเครื่องพิมพ์เอปซอนสามารถตั้งดิพสวิตช์เพื่อเลือกใช้หน่วยความจำภายในเครื่องพิมพ์ได้ เมื่อต้องการพิมพ์รหัสแถบให้ตั้งสวิตช์ในตำแหน่งเลือกใช้หน่วยความจำจะทำให้พิมพ์ได้เร็วขึ้นบ้างเล็กน้อย

หวังว่าผู้อ่านคงจะได้รับประโยชน์จากการนำโปรแกรม และหลักการนี้ไปดัดแปลงประยุกต์ใช้งานได้โดยไม่ต้องซื้อหาจากต่างประเทศ ถ้ามีโอกาสจะนำหลักการของเครื่องอ่านรหัสแถบและวงจรที่ทางศูนย์วิจัยและพัฒนางานคอมพิวเตอร์ได้ทดลองสร้างขึ้น โดยใช้งบประมาณไม่สูงมากนัก ตลอดจนโปรแกรมสำหรับควบคุมการอ่านเพื่อประมวลผลมาเสนอต่อไป

```
400 '**** INPUT DATA SEQUENCE FOR 3 OF 9 ****"
    Q$ = ""
    CLS
    LOCATE 5,5  : PRINT "INPUT DATA SEQUENCE OF 3 OF 9 "
    LOCATE 7,5  : PRINT "ALPHANUMERIC, CAPITAL LETTER ONLY"
    LOCATE 9,5  : PRINT "NOT LONGER THAN 20 CHARACTERS FOR STANDARD FORMAT"
    LOCATE 11,5 : PRINT "AND NOT LONGER THAN 30 CHARACTERS FOR COMPACT FORMAT"
    LOCATE 13,5 : INPUT "ENTER DATA SEQUENCE : ";Q$
    IF LEN(Q$) > 20 AND C1 = 3 OR LEN(Q$) > 30 AND C1 = 2 THEN
        PRINT CHR$(7)
        LOCATE 15,5 : PRINT "DATA TO LONG, REENTER"
        LOCATE 17,5 : PRINT "PRESS ANY KEY .......";
        Q$ = INPUT$(1)
        GOTO 400
    END IF
    T$ = ""
    LOCATE 15,5 : PRINT "3 OF 9 BAR CODE"
    FOR I = 1 TO LEN(Q$)
        T$ = T$+D$(ASC(MID$(Q$,I,1))-31)+"0"
    NEXT I
    T$ = D$(11)+"0"+T$+D$(11)
    FOR I = 1 TO LEN(T$)
        R(I)=VAL(MID$(T$,I,1))
    NEXT I
    PRINT #1, "3 OF 9"
    PRINT #1,
    GOSUB 600
    RETURN
600 '**** PRINT BAR CODE ****
    N1 = LEN(T$)
    K  = 0
    N6 = 0
    FOR I = 1 TO LEN(T$)
        N6 = N6+R(I)*C2+C1
    NEXT I
    N4 = FIX(N6/256)
    N8 = ((N6/256-N4)*256)
    FOR M1 = 1 TO M
        PRINT #1, CHR$(27);"3";CHR$(5);
        PRINT #1, CHR$(27);"*";CHR$(1);CHR$(N8);CHR$(N4);
        K=0
        FOR J = 1 TO N1
            J2 = R(J)*C2+C1
            IF K = 0 THEN K = 1 ELSE K = 0
            FOR I = 1 TO J2
                PRINT #1, CHR$(127*K);
            NEXT I
        NEXT J
        PRINT #1,
        PRINT #1, CHR$(27);"@";
    NEXT M1
    PRINT #1, CHR$(27);"@"
    IF P$ = "2" THEN
        PRINT #1, CHR$(27);CHR$(14);"* ";Q$;" *"
    ELSE
        PRINT #1, CHR$(27);CHR$(14);Q$
    END IF
    RETURN
```